Boolean Grammars

Rémi Varloot

25 December 2010

Boolean grammars are a generalization of context-free grammars in which one can include new logical operators when defining rules. Standard context-free grammars already allow us to use the union operator, where multiple rules can begin with the same nonterminal symbol. Boolean grammars extend this concept by introducing the set intersection and negation operators in their rules. We shall see that this generalization comes at a cost, however, as not all such grammars yield a unique solution.

In order to properly introduce Boolean grammars, we first define the notion of systems of equations for languages. Boolean grammars are then properly introduced. The last part of this paper takes a look at a means of properly defining a unique language attached to each grammar: naturally reachable solutions.

Contents

1 Language equations

This section is aimed at properly defining systems of equations, which are very useful when dealing with Boolean grammars.

First, we introduce *language formulae*:

Definition 1. Language formulae *over an alphabet* Σ *in variables* $X = (X_1, \dots X_n)$ *are defined inductively as follows:*

- *the empty word* ϵ *is a formula;*
- *a symbol from* Σ *is a formula;*
- *a variable from* X *is a formula;*
- *if* ϕ *and* ψ *are formulae, then:*
  - *the concatenation* $\phi \cdot \psi$ *is a formula,*
  - *the union* $\phi \vee \psi$ *is a formula,*
  - *the intersection* $\phi \ \& \ \psi$ *is a formula,*
  - *the negation* $\neg\phi$ *is a formula.*

The default precedence of operators is, from highest precedence to lowest: the concatenation ($\cdot$), the negation ($\neg$), the intersection (&) and the union ($\vee$).

We can now go on to define the *value of a formula*:

Definition 2. *The* value of a formula ϕ *over a vector of languages* $L = (L_1, \dots, L_n)$*, denoted as* $\phi(L)$*, is defined inductively as follows:*

- $\epsilon(L) = \{\epsilon\}$;
- $\forall a \in \Sigma, a(L) = \{a\}$;
- $\forall X_i \in X, X_i(L) = \{L_i\}$;
- $\forall(\phi, \psi)$:
  - $\phi \cdot \psi(L) = \phi(L) \cdot \psi(L)$,
  - $\phi \vee \psi(L) = \phi(L) \cup \psi(L)$,
  - $\phi \ \& \ \psi(L) = \phi(L) \cap \psi(L)$,
  - $\neg\phi(L) = \Sigma^* \setminus \psi(L)$.[1]

[1] $\Sigma^* = \{w \mid \exists n > 0, \exists(u_1, \dots, u_n) \in \Sigma^n, w = u_1 \cdot \dots \cdot u_n\} \cup \{\epsilon\}$

We finish by introducing *resolved systems of equations*:

Definition 3. *A* resolved systems of equations over an alphabet Σ in variables X *is a system*

$$X = \phi(X) \quad = \quad \begin{cases} X_1 & = & \phi_1(X) \\ & \vdots & \\ X_n & = & \phi_n(X) \end{cases}$$

where $\phi = (\phi_1, \ldots, \phi_n)$ *is a vector of formulae over* Σ *in variables* X*.*

Any vector of languages L *is said to be a* solution *of this system if* $L = \phi(L)$*.*

This definition established, we now illustrate how language formulae can be used to define languages:

Example 1. *Consider the following system of equations over the alphabet* $\{a, b\}$ *in variables* (X_1, X_2, X_3, X_4)*:*

$$\begin{cases} X_1 & = & \neg X_2 \cdot X_3 \ \& \ \neg X_3 \cdot X_2 \ \& \ X_4 \\ X_2 & = & (a \vee b) \cdot X_2 \cdot (a \vee b) \vee a \\ X_3 & = & (a \vee b) \cdot X_3 \cdot (a \vee b) \vee b \\ X_4 & = & (a \cdot a \vee a \cdot b \vee b \cdot a \vee b \cdot b) \cdot X_4 \vee \epsilon \end{cases}$$

This system has a unique solution L*, and we have:*

$$L_1 = \{w \cdot w \mid w \in \{a, b\}^*\}.$$

Though the resolution of this system shall not be demonstrated here, it is important to underline that there is no known denotation of this language using normal context-free languages or conjunctive languages.[2]

Though defining languages by means of such systems — as the first component of the solution of a system — would be most satisfactory, this method is not applicable as *not all systems yield a unique solution*, and as it has furthermore been proven that the class of these languages is the same as that of recursive sets, which is simply too big. We introduce Boolean grammars, but keep in mind that a proper characterization of a unique solution to such systems is necessary for these grammars to be correctly defined.

[2] *Conjunctive languages* are a restricted form of Boolean grammars, where only the intersection operator — not the negation — has been added to standard context-free grammars.

2 Boolean grammars

We are now ready to define Boolean grammars.

Definition 4. *A Boolean grammar is a quadruple $G = (\Sigma, N, P, S)$ in which:*

- *Σ is a finite nonempty set of terminal symbols;*
- *N is a finite nonempty set of nonterminal symbols, with $N \cap \Sigma = \varnothing$;*
- *P is a finite set of rules of the form*

$$A \to \alpha_1 \ \& \ \ldots \ \& \ \alpha_k \ \& \ \neg\alpha_{k+1} \ \& \ \ldots \ \& \ \neg\alpha_{k+l}$$

  where $k + l > 0$ and $\alpha_i \in (\Sigma \cup N)^*$ for all i in $\{1, \ldots, k+l\}$;
- *$S \in N$ is the start symbol of the grammar.*

The grammar is then interpreted as the following system:

$$\begin{cases} S &= \bigvee_{\phi \in \{\psi \mid S \to \psi \in P\}} \phi \\ A_1 &= \bigvee_{\phi \in \{\psi \mid A_1 \to \psi \in P\}} \phi \\ &\vdots \\ A_n &= \bigvee_{\phi \in \{\psi \mid A_n \to \psi \in P\}} \phi \end{cases} \qquad \text{for } N = \{S, A_1, \ldots A_n\}$$

A language L is generated by the grammar if $L = S(L')$ where L' is a solution of the system.

We give an example of a Boolean grammar.

Example 2. *Let us consider the same language as before:*

$$L = \{w \cdot w \mid w \in \{a, b\}^*\}.$$

A corresponding grammar would be

$$G = (\{a, b\}, \{S, A, B, C, X\}, P, S)$$

where P is the following set of rules:

$$S \to \neg A \cdot B \ \& \ \neg B \cdot A \ \& \ C \qquad C \to X \cdot X \cdot C \qquad C \to \epsilon$$

$$A \to X \cdot A \cdot X \qquad A \to a \qquad X \to a$$

$$B \to X \cdot B \cdot X \qquad B \to b \qquad X \to b$$

Furthermore, L is the unique language generated by this grammar.

Though Boolean grammars appear to be an adequate means of defining languages, it is however important to remind that not all systems yield a unique solution, and that this definition is therefore not yet satisfactory. We now give a means of properly characterizing unique solutions.

3 Characterization of systems with naturally reachable solutions

There are different ways to introduce a semantic for a proper solution. The one given here is that of *naturally reachable solutions.*

We begin by defining modulo equality and closure under substring for languages:

Definition 5. *Let L_1 and L_2 be two languages on Σ and $M \subseteq \Sigma^*$. L_1 and L_2 are* equal modulo M *if $L_1 \cap M = L_2 \cap M$. We denote this $L_1 = L_2 \pmod{M}$.*

Definition 6. *A language L is* closed under substring *if, for each word w in L, every substring of w is in L.*

We now have the appropriate tools for defining a naturally reachable solution:

Definition 7. *A vector of languages $L = (L_1, \ldots, L_n)$ is called a* naturally reachable solution *of a system $X = \phi(X)$ if:*

- *for every finite modulus M closed under substring,*
- *for every string $u \notin M$ such that all of u's proper substrings are in M,*
- *for every sequence $(\sigma_i)_{i \in \mathbf{N}}$ in $\{1, \ldots, n\}^*$,*[3]

the sequence $\left(L^{(i)}\right)_{i \in \mathbf{N}}$ defined by:

$$L^{(1)} = (L_1 \cap M, \ldots, L_n \cap M)$$

$$\forall i \in \mathbf{N}, \forall j \in \{1, \ldots, n\} \qquad L_j^{(i+1)} = \begin{cases} \phi_j\left(L^{(i)}\right) \cap (M \cup \{u\}) & \text{if } \sigma_i = j \\ L_j^{(i)} & \text{otherwise} \end{cases}$$

converges to

$$\left(L_1 \cap (M \cup \{u\}), \ldots, L_n \cap (M \cup \{u\})\right)$$

in a finite number of steps.

With this new definition, we have access to a new class of languages: naturally reachable solutions of systems of equations.

Furthermore, we have he following theorem, which makes these languages all the more interesting:

[3] $\mathbf{N}$ denotes the set of positive integers, that is to say $\{1, 2, \ldots\}$.

Theorem 1. *A naturally reachable solution is a solution, and no system can have more than one such solution.*

In order to properly prove this theorem, two intermediate results must first be established:

Proposition 1. *If two languages L' and L'' are not equal, then there exists a finite language M closed under substring such that $L' \neq L'' \pmod{M}$.*

Proof. $L' \neq L''$ means there there exists a substring w in $L' \Delta L''$. We conclude by noticing that $L' \neq L'' \left(\mathrm{mod}\, substrings\left(w\right)\right)$. □

Lemma 1. *If a vector of languages L is a solution of a given system $X = \phi\left(X\right)$ modulo every finite language M closed under substring, then L is a solution of the system.*

Proof. Suppose L is not a solution of the system. The previous proposition immediately yields the existence of a finite language M closed under substring such that $L \neq \phi\left(L\right) \pmod{M}$. □

Proof of the theorem. A naturally reachable solution being a solution modulo every finite language closed under substring, the lemma tells us that it is a solution.

Let us now suppose two such solutions L_1 and L_2 exist. We can prove that they are equal modulo all finite languages M closed under substring inductively on $|M|$.

Clearly, $L_1 = L_2 \left(\mathrm{mod}\, \varnothing\right)$. Furthermore, supposing that $L_1 = L_2 \left(\mathrm{mod}\, M\right)$, we have that

$$\left(L_1^{(i)}\right)_{i \in \mathbf{N}} \qquad \text{and} \qquad \left(L_2^{(i)}\right)_{i \in \mathbf{N}}$$

both converge towards the same sequence. In other words:

$$L_1 = L_2 \left(\mathrm{mod}\, M \cup \{u\}\right).$$

Hence the result, from which we can use the proposition to conclude that $L_1 = L_2$. □

We have proven that, for all system of equations, there is at most one unique naturally reachable solution. Though we lack a characterization of systems where such a solution exists, we can already offer the following definition:

Definition 8. *If the system corresponding to G has a naturally reachable solution L', then the language generated by a grammar G is the language $L\left(G\right) = S\left(L'\right)$.*

It is important to underline that this is *a* definition, and that it is not unique. Other means of characterizing systems with unique solutions or specific solutions for given systems exist, such as the *semantics of the unique solution in the strong sense*, and these can yield other definitions. What matters is defining which criterion will be used when introducing a Boolean grammar.

Bibliography

1. A. Okhotin, "Boolean grammars", *Information and Computation*, 194 (2004) 19–48.

2. A. Okhotin, "Decision problems for language equations with Boolean operations", *Automata, Languages and Programming*, LNCS 2719 (2003), 239–251.